This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Advancing FaaS Applications in the Cloud Continuum

Presentation for Red Hat Research Days -- Nov 16, 2022

Main Speaker: Georgios Kousiouris (HUA)
Conversation Leader: Luis Tomás Bolívar (RHT) and Yiannis Georgiou (RYAX)
This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047

AGENDA

❑ Introduction to EU Projects
❑ PHYSICS EU Project
  ○ Consortium, WPs
  ○ Objectives
  ○ Components
  ○ Research topics and results
❑ Red Hat role and main technologies
❑ K8s Image Layer Scheduler: benefits of open source
INTRODUCTION TO EU PROJECTS
3 main types

❖ Research and Innovation Actions (RIA)
  ❖ Typically aiming to deliver prototypes and integrated environments (TRL 3-4-5)

❖ Innovation Actions (IA)
  ❖ Start from existing prototypes (TRL3-4) and increase the TRL or target initial commercial services

❖ CSA actions
  ❖ Support actions aiming at standardization, organization of dissemination etc.

❖ Budget Sizes: depends on the EC call for proposals
EC calls

- Centered around 7-year Framework Programs

- Work programmes of 1-2 years ahead issued by the EC

- Consortia of partners organize and target a specific call:
  - Call has a specific high level scope: Smart cloud/edge continuum
  - Balance between academic and industrial partners (both large players and SMEs)
  - Balance between activities: R&D (if a RIA), piloting applications, innovation, dissemination, exploitation

- Work is split into Work Packages (WPs) and tasks inside WPs
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 101017047
Challenges targeted by PHYSICS

- Abstract usage of service offerings and clusters across the Continuum
- Adaptation of code to new serverless computing paradigms
- Investigation of space (location of execution)-time (duration of execution) in the continuum
- Optimization of resource selection and operation (global and local level)
- Multiple Exploitation Channels and Reusable Artefacts
Project Goals

Visual programming environment to create serverless workflows with reusable patterns and increased semantics

Platform-level functionalities to orchestrate and deploy FaaS workflows and optimize cloud/edge interplay

Provider-local resource management mechanisms to offer competitive and optimized services execution

This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047
This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047

- H2020 ICT 40 - Cloud Computing: towards a smart cloud computing continuum
- Research & Innovation Action
- Project Budget: 4,985,712€
- EC Grant (100%): 4,985,712€
- Timeframe: 01-01-2021 to 31-12-2023
- 14 Partners
- GFT is Project Coordinator

<table>
<thead>
<tr>
<th>1. GFT – Italy</th>
<th>8. INQBIT – Romania</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. ATOS IT – Spain</td>
<td>9. INNOVATION SPRINT – Belgium</td>
</tr>
<tr>
<td>3. HPE – Italy</td>
<td>10. INNOV-ACTS – Cyprus</td>
</tr>
<tr>
<td>4. RED HAT – Israel</td>
<td>11. CYBELETECH – France</td>
</tr>
<tr>
<td>5. FUJITSU – Germany</td>
<td>12. UPM – Spain</td>
</tr>
<tr>
<td>6. BYTE COMPUTER – Greece</td>
<td>13. HUA – Greece</td>
</tr>
<tr>
<td>7. RYAX TECHNOLOGIES - France</td>
<td>14. DFKI - Germany</td>
</tr>
</tbody>
</table>
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 101017047

**High Level Project Plan**

<table>
<thead>
<tr>
<th>M1-M7</th>
<th>M8-M18</th>
<th>M19-M23</th>
<th>M24-M34</th>
<th>M35-M36</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PHASE 1</strong>&lt;br&gt;DESIGN&lt;br&gt;• Market gap analysis&lt;br&gt;• Requirements gathering&lt;br&gt;• Overall architecture&lt;br&gt;• UC scenarios</td>
<td><strong>PHASE 2</strong>&lt;br&gt;IMPLEMENTATION (1&lt;sup&gt;st&lt;/sup&gt; ITERATION)&lt;br&gt;• Core features and implementation development&lt;br&gt;• Internal UC validation</td>
<td><strong>PHASE 3</strong>&lt;br&gt;1&lt;sup&gt;st&lt;/sup&gt; IMPACT&lt;br&gt;• Collect feedback from external audiences including EC</td>
<td><strong>PHASE 4</strong>&lt;br&gt;IMPLEMENTATION (2&lt;sup&gt;nd&lt;/sup&gt; ITERATION)&lt;br&gt;• final implementation round targeting the necessary extended functionalities</td>
<td><strong>PHASE 5</strong>&lt;br&gt;2&lt;sup&gt;nd&lt;/sup&gt; IMPACT&lt;br&gt;• Validation, usability, business evaluation and impact creation</td>
</tr>
</tbody>
</table>

**PHASE 1**

- DESIGN
  - Market gap analysis
  - Requirements gathering
  - Overall architecture
  - UC scenarios

**PHASE 2**

- IMPLEMENTATION (1<sup>st</sup> ITERATION)
  - Core features and implementation development
  - Internal UC validation

**PHASE 3**

- 1<sup>st</sup> IMPACT
  - Collect feedback from external audiences including EC

**PHASE 4**

- IMPLEMENTATION (2<sup>nd</sup> ITERATION)
  - final implementation round targeting the necessary extended functionalities

**PHASE 5**

- 2<sup>nd</sup> IMPACT
  - Validation, usability, business evaluation and impact creation
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 101017047

WPs

- WP2: Requirements and Architecture
- WP3: User Layer
  - Design Environment & Function DevOps
  - Reusable Patterns, semantics and handles from WP4/5
- WP4: Platform Services
  - Semantics Management
  - Placement Optimization
  - Performance Analysis
  - Data Services
  - Deployment Orchestration Services
- WP5: Infrastructure Services
  - Resource Controllers & Semantics
  - Local placement optimizers (container node selection)
- WP6: Applications, Pilots and Platform Integration

WP1 Project Management and Administration

WP2

WP3

WP4

WP5

Applications Design

Deployment Specs

App graph & semantics

APIs and resource semantics

Deployed Artefacts

Feedback

WP6

WP7 Exploitation, Dissemination and Impact Creation
WP3: User Layer
- Design Environment & Function DevOps
- Reusable Patterns, semantics and handles (e.g. controllers) from WP4/5

WP4: Platform Services
- Semantics Management
- Placement Optimization
- Performance Analysis
- Data Services
- Deployment Orchestration Services

WP5: Infrastructure Services
- Resource Controllers & Semantics
- Local placement optimizers (container node selection)
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047.
PHYSICS Design Environment

Joint Work from GFT, HPE and HUA
Goals of PHYSICS DE: Enhance, Abstract, Enrich FaaS application creation

- Customize function environment
- Simplify the creation of complex function workflows
  - Support complex primitives like Joins
- Exploit reusable patterns
- Increase semantics
- Abstract packaging, testing and deployment

<table>
<thead>
<tr>
<th>Case</th>
<th>Measured Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of errors</td>
<td>10</td>
</tr>
<tr>
<td>Errors solved with Node-RED test</td>
<td>7</td>
</tr>
<tr>
<td>Errors needing FaaS deployment</td>
<td>3</td>
</tr>
<tr>
<td>Local Node-RED test</td>
<td>1-2 seconds</td>
</tr>
<tr>
<td>FaaS deployment and test (including image creation)</td>
<td>5 minutes</td>
</tr>
<tr>
<td>Time saved</td>
<td>35 minutes</td>
</tr>
</tbody>
</table>
Baseline Technologies

- **Node-RED**
  - Programming environment for event driven applications
  - Built-in nodes
  - NPM extension nodes
  - Subflows (groups of functions) as nodes
  - Combined workflow orchestration and function execution abilities
  - Used as the main execution **runtime** and function **choreographer** for PHYSICS functions

- **Openwhisk**
  - Open source FaaS platform
  - Supported by IBM
  - Main tool behind IBM Cloud Functions Service
  - Can execute functions based on custom container images
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 101017047
PHYSICS Design Environment Backend

This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047
Design Environment Demo

- Full video available at: https://www.youtube.com/watch?v=DO2sFdfCD-o

- Function creation
- Function Build Management
- Function Testing
- Logging
- App creation
Embedded Node Red with enriched PHYSICS patterns and annotators - used to create and annotate flows
View and build list of available functions, created in Node Red tool.

<table>
<thead>
<tr>
<th>Document ID</th>
<th>URL</th>
<th>Action name</th>
<th>Built date</th>
</tr>
</thead>
<tbody>
<tr>
<td>62b972a9f245128e087ca33f</td>
<td>registry.apps.ocphub.physics-faaS.eu/custom/master:181</td>
<td>New_test_flow_master_0b0db467-a484-4023-9acc-626931e3afe5.json</td>
<td>6/27/22, 11:04 AM</td>
</tr>
</tbody>
</table>

This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047
Invoke OpenWhisk functions built by user or for other branches

Setting parameters for invoke

Parameter key * first
Parameter value * test value

Parameter key * second
Parameter value * other value

Invoke

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047
Example view for the result of function invocation.

Most recent results

Hello_DMS_george_a763502c-95ae-4f88-91d3-047a2b02d4ba.json

070f49fc6ed74da48f49fc6ed72da488

<table>
<thead>
<tr>
<th>Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>{</td>
</tr>
</tbody>
</table>
|   "data": "hello Blazej",
|   "msgResult": { |
|     "error": "The action did not return a dictionary." |
| }      |
| }      |
App creation from selected graphs: package and manage multiple functions as one application for deployment
Created Draft view

Consists of the flows that are already built

and

Flows for which had been triggered built process.
Displaying logs from microservices with useful information

- **timestamp**
- **microservice name**
- **client ID**
- **source class**
- **message**

Optional payload as JSON with button for easier copying.
User repository synchronization tool

View for checking the status

Result of the synchronization of the repository

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 101017047
Deployment status

Button for starting the deployment

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047
Patterns to support reusability, manageability and abstracted functionality

- Parallelization
- Context Management
- Data Collection
- Request Management
- Workflow primitives
- Etc.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 101017047.
Example of Fork-Join parallelization pattern

Concept

Object Array Input

Split Function  FaaS Invoker  Join Function


Implementation

Parameterization

Usage in a Function

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 101017047
Available semantic nodes

- Function relative placement (Affinity, Antiaffinity)
- Function Locality (e.g. Cloud/Edge)
  - Dynamically retrieves available clusters
- Importance (High/Medium/Low)
- OptimizationGoal (Performance, Energy etc)
- Function Sizing
- CPU Architecture requirements
- Intrafunction monitoring
- DMS interface for linking to data services
  - Can indicate data locality
Node-RED specific artefacts

- Pattern implementations as Node-RED reusable subflows
  - Exploitable as individual artefacts
  - Inclusion in Node-RED repo with documentation
    - [https://flows.nodered.org/collection/HXSkA2JILcG](https://flows.nodered.org/collection/HXSkA2JILcG)
  - Included in PHYSICS RAMP as artefacts along with other PHYSICS components

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 101017047.
PHYSICS RAMP

- Reusable Artefacts Marketplace Platform

- Catalogue of available PHYSICS artefacts
  - Operators, Node-RED flows, Datasets, Services, Semantic Models
  - Links to repositories and usage guides
  - Ability to download and use
    - **Looking for testers from the community!**
  - Ability to request a specific asset or an extension to an asset
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

PHYSICS Performance Aspects
**Function Choreography Overheads in Different Modes**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Warm Function Sequence Execution (ms)</th>
<th>Cold Penalty (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OW-OW</td>
<td>$T_{\text{total}} = 2.11 + n \times (T_{\text{function delay}} + 118.94)$</td>
<td>2248</td>
</tr>
<tr>
<td>NR-NR</td>
<td>$T_{\text{total}} = 128.36 + n \times (T_{\text{function delay}} + 1.69)$</td>
<td>7373</td>
</tr>
<tr>
<td>OW-NR</td>
<td>$T_{\text{total}} = 114.08 + n \times (T_{\text{function delay}} + 121.70)$</td>
<td>9621</td>
</tr>
</tbody>
</table>

**Sequences Only**

**Full primitives/Intra-container Parallelization**

**Diagram:**

- **a) OW-OW**
  - Target Workflow:
    - function 1
    - function 2
    - function 3
  - Node.js container
  - OpenWhisk Interface & Sequence Operator
  - External Trigger

- **b) NR-NR**
  - Target Workflow:
    - function 1
    - function 2
    - function 3
  - Node-RED custom container
  - OpenWhisk Interface
  - External Trigger

- **c) OW-NR**
  - Target Workflow:
    - function 1
    - function 2
    - function 3
  - Node-RED runtime
  - OW client
  - External Trigger

---

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 101017047.
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047
PHYSICS Load Generator Metrics

This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047
Space time continuum measurements

- Identification of
  - Function sizing considerations
  - Main cluster features
    - As interpreted from relative execution time, wait time, latency
  - Misconfigurations
  - Concurrent containers effect

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Workers</th>
<th>Worker Size</th>
<th>Container Pool Memory per Node</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>HUA</td>
<td>1</td>
<td>4 vCPU-8 GB RAM</td>
<td>8192 MB</td>
<td>Greece</td>
</tr>
<tr>
<td>AWS</td>
<td>4</td>
<td>4 vCPU-16 GB RAM</td>
<td>8192 MB</td>
<td>Sweden</td>
</tr>
<tr>
<td>Azure</td>
<td>3</td>
<td>8 vCPU-32 GB RAM</td>
<td>2048 MB</td>
<td>Netherlands</td>
</tr>
</tbody>
</table>
This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047

Red Hat Role and Main Technologies
Red Hat Main Role

- Infrastructure
  - Lower layers of the stack
    - Selection of infrastructure components: K8s, multicloud orchestration & networking
  - Infrastructure and APIs extensions
- Integration
  - API definitions (CRDs)
  - Kubernetes Operators
  - Kubernetes Webhooks
Technologies

- Kubernetes as the core (also OKD, OpenShift Community version)
- Submariner and Open Cluster Management (OCM) for multicluster setup
- MicroShift for low-footprint k8s devices
- Prometheus for monitoring data (both infrastructure and applications)
- OpenWhisk for Function as a Service
- K8s Operators and Webhooks
  - Kubernetes way to manage applications/components running on top of Kubernetes
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047
This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047.
This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047
K8s Image Layer Scheduling: Benefits of Open Source
Two-level scheduling for the continuum

Proposing a two-level scheduling procedure for the placement of the application components to available and suitable candidate cloud and Edge resources:

- **1st Scheduling Level - Global** allows the selection of most adapted cluster considering aspects such as performance, energy, etc.

- **2nd Scheduling Level - Local** enables the selection of most adapted node considering FaaS related optimizations, etc.

This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047.
1st level - Global Continuum Placement

- Implemented our first version of Global Continuum Placement component:
  - as a multi-objective scheduler allowing multiple objectives such as: Performance, Energy, etc
  - allowing the expression of both constraints and resources needs
  - the code is published as open-source: https://github.com/RyaxTech/global-continuum-placement

- Currently working on multi-objective optimizations based on linear programming
  - trying to minimize makespan and the amount of downloaded data by containers and function’s I/O while considering platform heterogeneity
  - adaptable to various objectives
  - an article is being prepared to be submitted to CCGrid-2023
2nd level - Local Adaptive Scheduling

- Focus in intelligent scheduling algorithms for efficient resource sharing (CPU, memory, storage, network), load balancing and managing resources for FaaS execution.

- Based on the results of our studies using simulations and observations
  - Our first algorithm aims to minimize the delays due to image downloading for function execution: **minimizing Cold starts**
  - Kubernetes already provides an ImageLocality plugin
  - We have implemented a variation of ImageLocality taking into account the existence of Containers’ Layers and trying to favor the execution of functions on nodes where layers of the containers to be deployed exist already. The new scheduler is named **LayersLocality**

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 101017047
Layers Locality Scheduling Implementation

❖ For this implementation we need the following:
  ❖ To get the available layers on each node (name and size)
  ❖ For each new pod compute a score per node considering the cumulative size of already available layers

❖ For this we had to make changes on various areas of the different involved software without breaking retro-compatibility:
  ❖ Kubernetes internal interfaces to add the Layers info into Kubernetes: optional Layers field in the core.v1.Node.NodeStatus API
  ❖ Container Runtime CRI-O adapted to get available layers name and size on node and send it through annotations (no API change)
Towards pushing Layers Locality Scheduling in upstream K8S

- Open-source code and installation documentation available online on github
  https://github.com/RyaxTech/k8s-container-layer-locality
  https://github.com/RyaxTech/kubernetes
  https://github.com/RyaxTech/cri-o

- Discussions have been started with “sig-scheduling” Kubernetes group to push the new “Layers Locality” scheduler to the upstream Kubernetes.

- Currently working on experiments to compare the new scheduler with the typical Kubernetes scheduler and show the performance improvement for FaaS applications.
Thank you for the attention!

www.physics-faas.eu
linkedin.com/company/physicsh2020
https://twitter.com/H2020Physics
info@physics-faas.eu